

What is claimed is:

1 1. A method comprising:

2 producing in a first engine thread included in a
3 processor that processes packets, a vector that
4 represents content of a packet; and
5 storing the vector in a memory associated with the
6 processor such that the vector is accessible by a second
7 engine thread included in the processor.

1 2. The method of claim 1, wherein producing the vector
2 includes grouping and summing the content of the packet to
3 produce vector elements.

1 3. The method of claim 1, further comprising:

2 updating the vector on the second engine thread
3 included in the processor.

1 4. The method of claim 3, further comprising:

2 storing the updated vector in the memory such that
3 the updated vector is accessible by a third engine thread
4 included in the processor.

1 5. The method of claim 1, wherein the vector is representable
2 as $C = (C_1, C_2, \dots, C_n) = \left(\sum_i B_{i,1}, \sum_i B_{i,2}, \dots, \sum_i B_{i,n} \right)$.

1 6. The method of claim 3, wherein updating the vector
2 includes subtracting a byte from an element included in the
3 vector.

1 7. The method of claim 3, wherein updating the vector
2 includes rotating elements included in the vector.

1 8. The method of claim 3, wherein the updated vector is used
2 to determine a numerical value that represents the content of
3 the packet.

1 9. The method of claim 8, wherein the numerical value is
2 inserted into the packet.

1 10. The method of claim 8, wherein the numerical value is
2 used to verify payload of the packet.

1 11. A computer program product, tangibly embodied in an
2 information carrier, for monitoring content of a packet, the
3 computer program product being operable to cause a machine to:
4 produce in a first engine thread included in a
5 processor that processes packets, a vector that
6 represents content of the packet; and
7 store the vector in a memory associated with the
8 processor such that the vector is accessible by a second
9 engine thread included in the processor.

1 12. The computer program product of claim 11, wherein
2 producing the vector includes grouping and summing the content
3 of the packet to produce vector elements.

1 13. The computer program product of claim 11 being further
2 operable to cause a machine to:

3 update the vector on the second engine thread
4 included in the processor.

1 14. The computer program product of claim 13 being further
2 operable to cause a machine to:

3 store the updated vector in the memory such that the
4 updated vector is accessible by a third engine thread
5 included in the processor.

1 15. The computer program product of claim 11, wherein the
2 vector is representable as $C = (C_1, C_2, \dots, C_n) = \left(\sum_i B_{i,1}, \sum_i B_{i,2}, \dots, \sum_i B_{i,n} \right)$.

1 16. The computer program product of claim 13, wherein
2 updating the vector includes subtracting a byte from an
3 element included in the vector.

1 17. The computer program product of claim 13, wherein
2 updating the vector includes rotating elements included in the
3 vector.

1 18. The computer program product of claim 13, wherein the
2 updated vector is used to determine a numerical value that
3 represents the content of the packet.

1 19. The computer program product of claim 18, wherein the
2 numerical value is inserted into the packet.

1 20. The computer program product of claim 18, wherein the
2 numerical value is used to verify payload of the packet.

1 21. A packet verifier comprises:

2 a process to produce in a first engine thread
3 included in a processor that processes packets, a vector
4 that represents content of a packet; and
5 a process to store the vector in a memory associated
6 with the processor such that the vector is accessible by
7 a second engine thread included in the processor.

1 22. The packet verifier of claim 21, wherein producing the
2 vector includes grouping and summing the content of the packet
3 to produce vector elements.

1 23. The packet verifier of claim 21, further comprises:
2 a process to update the vector on the second engine
3 thread included in the processor.

1 24. The packet verifier of claim 23, further comprises:

2 a process to store the updated vector in the memory
3 such that the updated vector is accessible by a third
4 engine thread included in the processor.

1 25. The packet verifier of claim 21, wherein the vector is
2 representable as $C = (C_1, C_2, \dots, C_n) = \left(\sum_i B_{i,1}, \sum_i B_{i,2}, \dots, \sum_i B_{i,n} \right)$.

1 26. The packet verifier of claim 23, wherein updating the
2 vector includes subtracting a byte from an element included in
3 the vector.

1 27. The packet verifier of claim 23, wherein updating the
2 vector includes rotating elements included in the vector.

1 28. The packet verifier of claim 23, wherein the updated
2 vector is used to determine a numerical value that represents
3 the content of the packet.

1 29. The packet verifier of claim 28, wherein the numerical
2 value is inserted into the packet.

1 30. The packet verifier of claim 28, wherein the numerical
2 value is used to verify payload of the packet.

1 31. A system comprising:
2 a network processor capable of,

3 producing in a first engine thread included in
4 the network processor, a vector that represents
5 content of a packet, and
6 storing the vector in a memory associated with
7 the processor such that the vector is accessible by
8 a second engine thread included in the network
9 processor.

1 32. The system of claim 31, wherein producing the vector
2 includes grouping and summing the content of the packet to
3 produce vector elements.

1 33. The system of claim 31, wherein the network processor is
2 further capable of:

3 updating the vector on the second engine thread
4 included in the network processor.

1 34. A network forwarding device comprising:
2 an input port for receiving packets;
3 an output for delivering the received packets; and
4 a network processor capable of,
5 producing in a first engine thread included in
6 the network processor, a vector that represents
7 content of a packet, and
8 storing the vector in a memory associated with
9 the network processor such that the vector is

10 accessible by a second engine thread included in the
11 network processor.

1 35. The system of claim 34, wherein producing the vector
2 includes grouping and summing the content of the packet to
3 produce vector elements.

1 36. The system of claim 34, wherein the network processor is
2 further capable of updating the vector on the second engine
3 thread included in the network processor.

1 37. A method comprising:

2 grouping and summing content of a packet to produce
3 elements of a vector that represents the content of the
4 packet; and

5 storing the vector in a memory associated with a
6 processor such that the vector is accessible.

1 38. The method of claim 37, further comprising:

2 updating the vector to reflect processing of the
3 content of the packet.

1 39. The method of claim 37, wherein the vector is
2 representable as $C = (C_1, C_2, \dots, C_n) = \left(\sum_i B_{i,1}, \sum_i B_{i,2}, \dots, \sum_i B_{i,n} \right)$.